

# SIEMENS

Attorney Docket No. 2002P18325WOUS

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Inventor: M.-C Fritsch et al.	)	Group Art Unit:	2611
	)		
Serial No.: 10/538,152	)	Examiner:	SINGH, Hirdepal
	)		
Filed: 06-08-2005	)	Confirmation No.	3257
Title: SYSTEM FOR THE GENERATION OF CODE			

Mail Stop Appeal Brief - Patent  
COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

### APPELLANT'S BRIEF UNDER 37 CFR 41.37

This brief is in furtherance of the Notice of Appeal filed in this application on 08 August 2008 in response to the final office action of 08 April 2008.

#### 1. REAL PARTY IN INTEREST - 37 CFR 41.37(c)(1)(i)

The real party in interest in this Appeal is the assignee, Siemens Aktiengesellschaft.

#### 2. RELATED APPEALS AND INTERFERENCES - 37 CFR 41.37(c)(1)(ii)

There is no other appeal, interference or judicial proceeding that is related to or that will directly affect, or that will be directly affected by, or that will have a bearing on the Board's decision in this Appeal.

3. STATUS OF CLAIMS - 37 CFR 41.37(c)(1)(iii)

Claims pending: 13, 17, 19, 23, 26, 29, 31, 33, 34

Claims cancelled: 1-12, 14-16, 18, 20-22, 24, 25, 27, 28, 30, 32

Claims withdrawn but not cancelled: None

Claims allowed: None

Claims objected to: None

Claims rejected: 13, 17, 19, 23, 26, 29, 31, 33, 34

The claims on appeal are 13, 17, 19, 23, 26, 29, 31, 33, 34.

4. STATUS OF AMENDMENTS - 37 CFR 41.37(c)(1)(iv)

An amendment after final rejection was filed under 37 CFR 1.116 on 04 June 2008, and was entered.

5. SUMMARY OF THE CLAIMED SUBJECT MATTER- 37 CFR 41.37(c)(1)(v)

Applicants' page, line, and paragraph numbers mentioned herein are relative to the substitute specification. The line counts do not include blank lines.

This invention relates generally to a system for generating industrial automation code for a manufacturing and/or processing plant from a description with control-relevant information. The description is described in a drawing based on a material flow in the plant, and includes a structure of the plant, know-how, and predecessor/successor relationships between components in the description representing elements of the plant. FIG 1 schematically illustrates a system according to aspects of the invention. FIG 2 illustrates a structure of controllers in an automation system. FIG 3 illustrates an example of port/information mapping and generation of automation code according to aspects of the invention. The invention is partially summarized in paragraph 10 (page 3, line 23 to page 4, line 19). The independent claims, 13, 26, and 33, are annotated individually below, relative to the specification and drawings.

**Independent claim 13** is directed to a system (FIG 1) for generating automation code for a manufacturing and/or processing plant from a description (FIG 1, element 1; and page 3, line 23 to page 4 line 19) enriched with control-relevant information (page 3, lines 23-25; and page 9, lines 15-19. Control-relevant information is also called "automation-relevant information", as in page 8, lines 7-14), the system comprising:

components (2) in the description described in a drawing based on a material flow (page 5, line 26 to page 6, line 16) in the manufacturing and/or processing plant, wherein the drawing comprises control-relevant information (FIG 3 and page 11, lines 17-19), and the components have ports (FIG 1, element 6; and page 4, lines 3-9) and are represented by at least one functional module (FIG 1, element 3), wherein

input/output information is mapped to the ports, wherein the input/output information stems from directed relationships between the components (FIG 3 and page 11, lines 17-19), wherein the input/output information comprising predecessor/successor relationships among the components is included in the description (page 8 lines 19-24), wherein

signals (FIG 1, element 4; and page 8 lines 24-29) provided for a transmission via the ports of the components are associated with the functional module and further comprising:

a first mechanism (FIG 1 element 5; and page 8, lines 3-4) for defining metainformation (page 4, lines 9-23) for the signals; and

a code generator (FIG 1 element 7; and page 8, lines 5-7) for generating automation code by interconnecting the signals, wherein the automation code is generated on the basis of a structure of the plant (page 7, last two lines; and page 10, lines 13-16) and know-how, including the predecessor/successor relationships, previously input into the description (page 4, lines 12-16; and page 7, line 24 to page 8, line 6).

**Independent claim 26** is directed to a method for generating automation code for a manufacturing and/or processing plant from a description (FIG 1, element 1; and page 3, line 23 to page 4 line 19) enriched with control-relevant information (page 3, lines 23-25; and page 9, lines 15-19. Control-relevant information is also called "automation-relevant information", as in page 8, lines 7-14), the method comprising:

representing components described in the descriptions by at least one functional block (FIG 1, element 3) or building block in a drawing based on a material flow (page 5, line 26 to page 6, line 16) in the plant, wherein the drawing comprises control-relevant information (FIG 3 and page 11, lines 17-19), and each component has at least one port (FIG 1, element 6; and page 4, lines 3-9);

mapping input/output information regarding the ports between the components (FIG 3 and page 11, lines 17-19 ), wherein the input/output information stems from directed relationships including predecessor/successor relationships among the components contained in the descriptions (page 8 lines 19-24);

transmitting signals (FIG 1, element 4; and page 8 lines 24-29) associated with the functional blocks or building blocks via the ports of the components;

defining metainformation (page 4, lines 9-23) for the signals; and

generating automation code by interconnecting the signals (FIG 1 element 7; and page 8, lines 5-7), wherein the automation code is generated on the basis of a structure of the plant (page 7, last two lines; and page 10, lines 13-16) and know-how, including the predecessor/successor relationships, previously input into the description (page 4, lines 12-16; and page 7, line 24 to page 8, line 6).

**Independent claim 33** is directed to system (FIG 1) for generating automation code for a manufacturing and/or processing plant, the system comprising:

a plant description comprising a plurality of components, each component representing a given element of the plant (FIG 1, element 1; and page 3, line 23 to page 4 line 19), each component comprising at least one function module (FIG 1, element 3) and at least one port (FIG 1, element 6; and page 4, lines 3-9), each port representing a connection point on the given element for data communication with another element of the plant, each function module being a reusable software object type that defines characteristics and functions of the given element;

a communication network (FIG 2, element 10) within the plant comprising a respective controller (FIGs 2 and 3, element 11) connected to each of the plant elements;

the description comprising a drawing of the components based on a flow of material in the plant (page 5, line 26 to page 6, line 16) and control-relevant information (page 3, lines 23-25; and page 9, lines 15-19. Also called "automation-relevant information", as in page 8, lines 7-14) comprising rules that specify all allowable relationships (page 9, lines 1-5) including predecessor/successor relationships (page 8 lines 19-24; and FIG 3, page 11, lines 17-19) among the plant elements, including allowable information content and flow directions among the ports; and

a code generator (FIG 1 element 7; and page 8, lines 5-7) that automatically generates automation code for the plant that controls information flows among the controllers based on the description, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description (page 7, line 24 to page 8, line 6; page 9, lines 20-23; and page 10, lines 13-16).

6. GROUNDS OF REJECTION TO BE REVIEWED UPON APPEAL - 37 CFR  
41.37(c)(1)(vi)

Claims 13, 17, 19, 23, 26, 29, 31, 33, and 34 are rejected under 35 USC 103(a) as being unpatentable over Burgess (US 5,805,896), in view of Sakurai et al. (US 6,334,076), Kroeger (US 2002/0165723), and Elmquist ("A Uniform Architecture for Distributed Automation", Advances in Instrumentation and Control, Instrument Society of America, Research Triangle Park, NC US, Vol. 46, Part 2, 1991, Pages 1599-1608).

7. ARGUMENT 37 CFR 41.37(c)(1)(vii)

Arguments applicable to all claims:

The below arguments apply to all of the claims under appeal, including the independent claims 13, 26, and 33. Examiner rejected claims 13 and 26 in one section of the office action, and claim 33 in another section. However, the arguments below apply to elements and combinations common to all three of the independent claims 13, 26, and 33. Claim 33 does not use the term "directed relationships" as found in claims 13 and 26, but it recites "predecessor-successor relationships", which are directed relationships. Thus, the appeal board may select for review the claim they feel most clearly and particularly recites the invention.

I. Directed Relationships: The independent claims 13, 26, and 33 recite the generation of plant automation code by graphically mapping input/output information to ports on components based on information, including predecessor/successor relationships, contained in a description of the components. Burgess does not include predecessor/successor information in his component objects. Instead his directed relationships are defined at the graphical mapping stage by the programmer.

Examiner asserts on page 6 of the Office Action of 04-08-2008 that directed relationships of components are defined in Burgess col. 3, lines 29-34, lines 54-57 (FIG 4), and col. 4, lines 1-16 (FIG 5). However, these lines describe visual programming as illustrated in FIG 4, in which a programmer graphically configures the directed relationships. Since

predecessor/successor relationships are not contained in the descriptions of the objects 410, 420, 450, and 460, nothing prevents a reversal of the C-to-F and F-to-C calculator objects by the visual programmer, which would produce incorrect relationships.

Burgess col. 3, lines 21-38: FIG. 4 is a diagram illustrating visual programming of the present invention. To generate a visual program to implement a temperature converter, a programmer would position a Fahrenheit scroll bar 401, a Fahrenheit display 430, a Centigrade scroll bar 420, and a Centigrade display 440. The programmer would also position an FtoC calculator 460, which converts a Fahrenheit value to a Centigrade value, and a CtoF calculator 450, which converts a Centigrade value to a Fahrenheit value. In one embodiment, the components are selected from an extendible list of available components. The programmer then connects the components through their ports. The connections 412->461 and 412->431 indicate that when the Fahrenheit scroll bar is changed (e.g., slider moved), the new value is sent to the FtoC calculator and the Fahrenheit display. The connection 462->421 indicates that when the FtoC calculator calculates a new Centigrade value, the new value is sent to the Centigrade scroll bar.

In contrast, Applicants' directed relationships are already contained in a description of each component, constraining the connections to a proper order by allowing fewer degrees of freedom in order to reduce the possibility of error.

Applicants' paragraph 10, lines 1-3: "In the system according to the invention data continuity is achieved in that control-relevant information is already contained in a description."

Applicants' paragraph 20, lines 13-18: "The information already contained in the description 1 is used to allocate input and output information to the ports. The predecessor-successor relationships between the components are governed by this information, i.e. who sends data to whom via which data input is defined thereby."

This prevents incorrect relationships that are possible in Burgess, because the predecessor-successor relationships are defined prior to graphical mapping stage for local plant code generation.

Applicants' paragraph 21: "On the basis of the metainformation, the components 2 are connected to one another by automated means. Particular connections between the components 2 can only be implemented if this is permitted by the constraints described in the metainformation. Automated "wiring" of the components 2, and therefore automatic generation of automation code, are therefore effected."

Applicants directed relationships restrict freedom in order to reduce complexity in automation programming, because incorrect options are eliminated from consideration. Continuity of expert information and earlier know-how guides and limits the plant automation code developer.

Applicants' paragraph 22, all lines: "The work of the development engineer is greatly facilitated thereby, since fewer degrees of freedom exist as a result of the definition of the metainformation, reducing the possibilities of error. In addition, a continuous information flow is ensured, reducing the loss of already established know-how during the development of the automation system."

Applicants' paragraph 34, lines 2-3: automation code is generated on the basis of existing descriptions 1 of a plant structure.

II. Previously input plant structure and know-how: On top of page 7 of the office action, Examiner concedes that Burgess does not teach that (1) code generation for manufacturing or processing plants and automation code is generated on the basis of a structure of the plant and know-how previously input into the description. Know-how previously input into the description is recited in all of the independent claims. It includes predecessor/successor relationships, as taught in paragraphs 22-25 of the specification. This is not simply an intended use as Examiner asserts, but is a feature that prevents sequence errors in any use.

Sakurai provides standard program modules previously coded by program engineers. These standard modules are independent of plant type (col. 4, lines 3-6). A plant operator inputs a plant operation procedure by keyboard, which includes a time sequential control flow (col. 3, lines 61-63). This results in a selection of standard program source code modules for assembly into a load module for execution in a programmable logic controller (PLC). There is no teaching that the standard program modules include a previously entered description of predecessor/successor relationship for plant components.

III. Description in drawing: Examiner concedes on top of page 7 that (2) Burgess does not teach components described in a drawing comprising control-relevant information in a manufacturing and/or processing plant. Examiner then cites Sakurai col. 2, lines 20-51.

However, these lines do not mention a drawing or graphics at all. Examiner also cites Sakurai col. 4, lines 8-22. However, these lines describe visually monitoring of a PLC program while it is executing, in order to verify proper program operation -- not graphically configuring a specification for automatic code generation.

IV. Description based on material flow: Examiner concedes on top of page 7 that (3) Burgess does not teach control information described in a drawing based on a material flow in a manufacturing and/or processing plant. Examiner then asserts that Elmqvist supplies the claimed feature of drawings with control-relevant information based on material flow in a processing plant, because it is inherent that the physical objects of the plant form the path for the material or fluid flow as shown in the example of the tank system (figures 1-5). However, as in Burgess, this tank system layout only exists after the visual designer has selected the graphic components and placed them in this order. These graphical components do not have predecessor/successor descriptions stored in them to require an order based on a material flow and prevent mistakes. Instead, the design module definitions of Elmqvist are purely hierarchical (FIG 2). The first line under VISUALIZATION on page 1605 states: "The complete picture, as seen in a window, is a hierarchical picture according to the module hierarchy." The two tanks of FIG 1 are just instantiations of the same "tank" object. Thus their order in FIG 1 could be reversed -- the same kind of mistake as discussed for Burger above. It so happens that this reversal would not make a difference in FIG 1 of Elmqvist, but this is only because FIG 1 is too simple an example to illustrate an ordered relationship. The examples of Elmqvist and Burgess are both highly simplified, but at least the example of Burgess can be used to show how order is important, which is the case in a manufacturing or processing plant.

V. Predecessor/successor relationships: Examiner concedes on top of page 7 that (4) Burgess does not teach system components defined to have predecessor/successor relationships. Examiner then cites par. 112 of Kroeger. This paragraph describes a display of project management task records. However, these tasks may be edited/added at any time, thus changing their sequence (par. 113).

Kroeger par. 113: As set forth hereinabove, tasks may be edited/added at any time by authorized contacts. Further, the project task manager may provide an audit trail

of all task changes made, when they were made, and by whom. The project task manager may also display bar graphs and calendars of schedule tasks & relationships over a scrolling time period. As such, the project task manager may display real time projections of budgets and actual costs.

VI. Kroeger is incompatible: Kroeger teaches a project and document management system -- not a system for generating plant automation code or any kind of automation code. Examiner has not identified any elements in Kroeger that could correspond to manufacturing plant components. Kroeger simply coordinates tasks done by humans, which does not apply to the present invention. Kroger does not provide a graphical configuration process. FIGs 7 and 8 show a textual menu-driven screen. Kroeger provides full flexibility for changes in the schedule and contracts (the term "change order" is found 11 times in Kroeger). Flexibility is a primary purpose of Kroeger in order to overcome prior art inflexibility (par. 5, lines 1-5, and par. 6, lines 5-9). As such, this feature is not simply an alternate embodiment, and it is incompatible with the present invention, which reduces flexibility to avoid mistakes in plant automation code. This teaches away from the present invention, in which the predecessor/successor relationships are pre-established.

VII. M.P.E.P. 2143.03 provides that to establish *prima facie* obviousness of a claimed invention, all words in a claim must be considered in judging the patentability of that claim against the prior art. If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious.

MPEP 2106 II C:

" . . . Finally, when evaluating the scope of a claim, every limitation in the claim must be considered. USPTO personnel may not dissect a claimed invention into discrete elements and then evaluate the elements in isolation. Instead, the claim as a whole must be considered. See, e.g., *Diamond v. Diehr*, 450 U.S. 175, 188-89, 209 USPQ 1, 9 (1981) ("In determining the eligibility of respondents' claimed process for patent protection under § 101, their claims must be considered as a whole. It is inappropriate to dissect the claims into old and new elements and then to ignore the presence of the old elements in the analysis. This is particularly true in a process claim because a new combination of steps in a process may be patentable even though all the constituents of the combination were well known and in common use before the combination was made.")." "

As argued above, the proposed combination of Burger, Sakurai, Kroeger, and Elmquist does not teach or suggest several features of the present invention recited in the independent claims. These features provide major benefits in safety and continuity of plant automation design and code generation over the prior art. Furthermore, Kroeger is in a different field, and teaches away from the present invention, as argued above. Therefore, Applicant respectfully requests withdrawal of the 35 USC 103 rejections, and allowance of the present application.

8. CLAIMS APPENDIX - 37 CFR 41.37(c) (1) (viii).

A copy of the claims involved in this appeal is attached as a claims appendix under 37 CFR 41.37(c) (1) (viii).

9. EVIDENCE APPENDIX - 37 CFR 41.37(c) (1) (ix)

None is required under 37 CFR 41.37(c) (1) (ix).

10. RELATED PROCEEDINGS APPENDIX - 37 CFR 41.37(c) (1) (x)

None is required under 37 CFR 41.37(c) (1) (x).

Respectfully submitted,

Dated: 10/8/08

By: 

John P. Musone  
Registration No. 44,961  
(407) 736-6449

Siemens Corporation  
Intellectual Property Department  
170 Wood Avenue South  
Iselin, New Jersey 08830

## APPENDIX OF CLAIMS ON APPEAL

13. A system for generating automation code for a manufacturing and/or processing plant from a description enriched with control-relevant information, the system comprising:

components in the description described in a drawing based on a material flow in the manufacturing and/or processing plant, wherein the drawing comprises control-relevant information, and the components have ports and are represented by at least one functional module, wherein

input/output information is mapped to the ports, wherein the input/output information stems from directed relationships between the components, wherein the input/output information comprising predecessor/successor relationships among the components is included in the description, wherein

signals provided for a transmission via the ports of the components are associated with the functional module and further comprising:

a first mechanism for defining metainformation for the signals; and

a code generator for generating automation code by interconnecting the signals, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

17. The system according to claim 13, further comprising a mechanism for inputting control-relevant information for use in the description.

19. The system according to claim 13, wherein the material flow, and/or an energy flow, and/or an information flow in the plant is provided as a basis for mapping the directed relationships between the components.

23. The system according to claim 13, wherein the generation of automation code is provided for central and/or distributed automation solutions.

26. A method for generating automation code for a manufacturing and/or processing plant from at least one description enriched with control-relevant information, the method comprising:

representing components described in the descriptions by at least one functional block or building block in a drawing based on a material flow in the plant, wherein the drawing comprises control-relevant information, and each component has at least one port;

mapping input/output information regarding the ports between the components, wherein the input/output information stems from directed relationships including predecessor/successor relationships among the components contained in the descriptions;

transmitting signals associated with the functional blocks or building blocks via the ports of the components;

defining metainformation for the signals; and

generating automation code by interconnecting the signals, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

29. The method according to claim 26, wherein control-relevant information is input to the description.

31. The method according to claim 26, wherein automation code is generated for central and/or distributed automation systems.

33. A system for generating automation code for a manufacturing and/or processing plant, the system comprising:

a plant description comprising a plurality of components, each component representing a given element of the plant, each component comprising at least one function module and at least one port, each port representing a connection point on the given element for data communication with another element of the plant, each function module being a reusable software object type that defines characteristics and functions of the given element;

a communication network within the plant comprising a respective controller connected to each of the plant elements;

the description comprising a drawing of the components based on a flow of material in the plant and control-relevant information comprising rules that specify all allowable relationships including predecessor/successor relationships among the plant elements, including allowable information content and flow directions among the ports; and

a code generator that automatically generates automation code for the plant that controls information flows among the controllers based on the description, wherein the automation code is generated on the basis of a structure of the plant and know-how, including the predecessor/successor relationships, previously input into the description.

34. The system of claim 33, wherein the network comprises at least two control zones, each control zone comprising a subset of the plant elements, the network further comprises a coordinating controller for each control zone, and wherein the description describes a topology of the network for the automatic code generation.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.